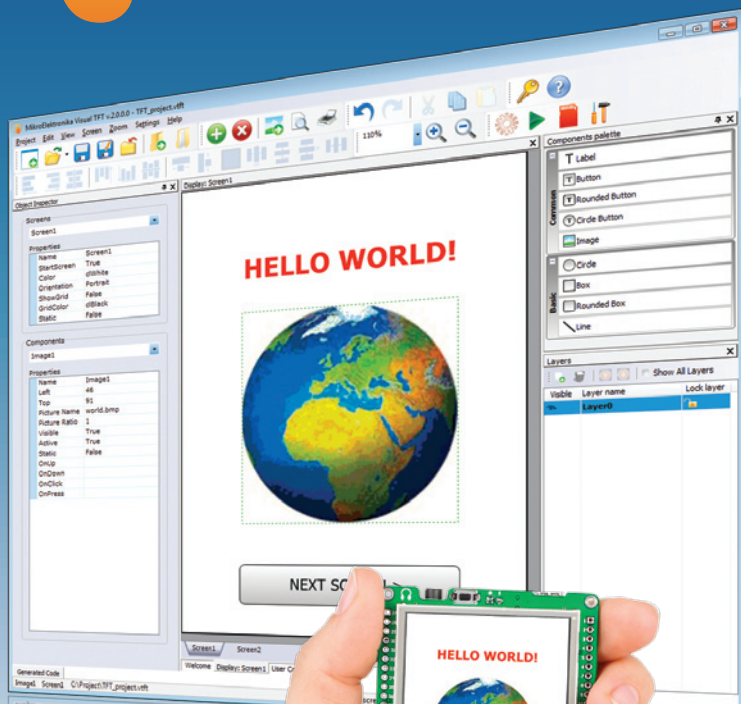


# Creating the first project in



# Visual TFT



This tutorial will show you how to make your first steps in Visual TFT software. Let's create a simple „Hello World“ application in less than two minutes.

# TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in MikroElektronika.

The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

A stylized, handwritten signature in black ink, consisting of a large 'C' followed by several loops and a long horizontal stroke.

Nebojsa Matic  
General Manager

# Table of Contents

---

Introduction to your first project	4
1. Creating a new project	5
Step 1 - Create a blank new project	5
Step 2 - Initial settings	7
2. Creating visual contents	9
Step 1 - Adding „Hello World“ label	10
Step 2 - Adding Earth image	12
Progress after image is added	14
Step 3 - Adding „NEXT SCREEN>“ button	15
Progress after button is added	17
3. Creating second screen	18
4. Writing code	19
5. Generating code	21
6. Start compiler	22
7. Build and program	23
8. The result	24

---

# Introduction to your first project

Let's start a journey into the world of Visual TFT software. Together we'll go through the series of easy steps into making our first fun „Hello world“ project. Along the way, we will learn how to add text, images and buttons, and invoke simple screen transitions with minimum programming. At the end of the first project design we will generate a code that can be built with your chosen compiler and uploaded to your target device.



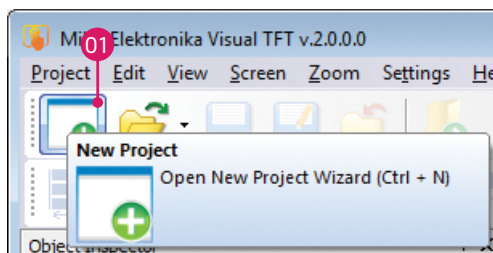
This is how our project should look like when it's completed and uploaded on target device

# 1. Creating a new project

Creating a new project is divided into a few simple steps. Follow them in order to complete this Hello World project.

## Step 1 - Create a blank new project

First, we have to create a new blank project and give it a name.



- 01 Click on a **New Project** button in upper left corner of Visual TFT window. New project window will appear.

Figure 1-1: New project

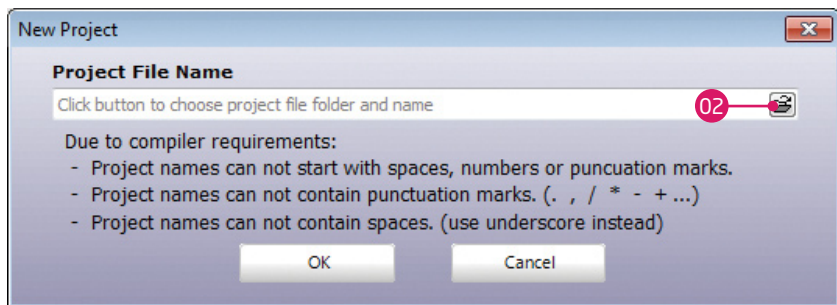
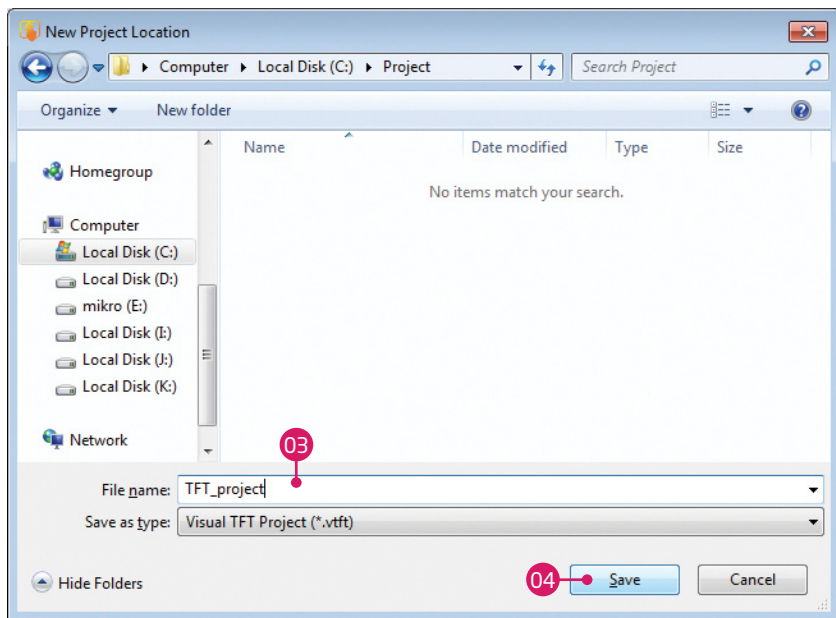


Figure 1-2: Project name

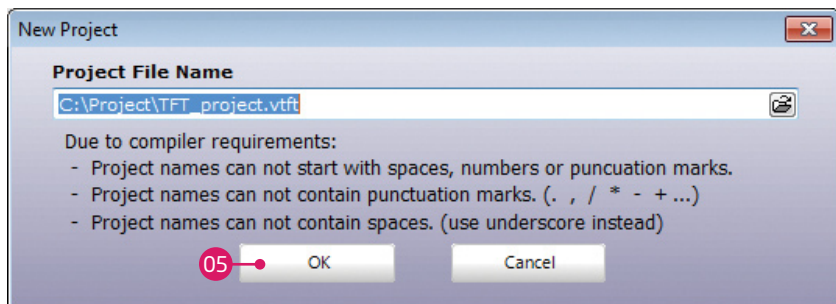
- 02 Click on open folder button

If you know exact path to your project location just type it to the **Project File Name** box.



**03** When **New Project Location Window** appears, select target folder for your project and type in the project name. For example: „TFT\_project“.

**04** Click **Save** button to proceed.



**05** Verify if everything is in order, and click OK button to create a blank new project.

## Step 2 - Initial settings

When creation of the blank new project is completed a new window with options will appear. We made all those complicated settings as easy as it gets. Just select the target hardware and compiler from the drop down list and everything will be automatically set for you.

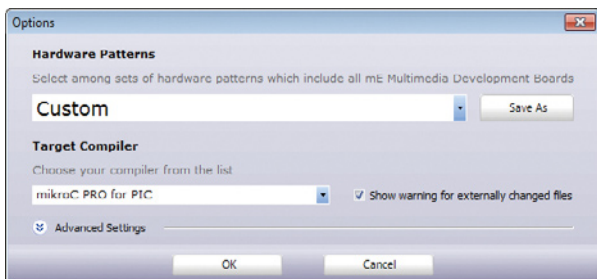
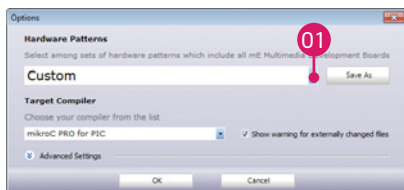


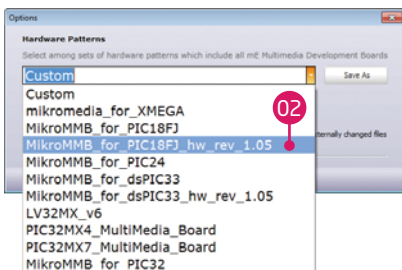
Figure 1-3: Options window

### Choose target hardware

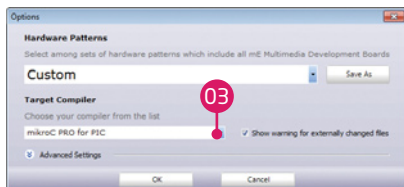


01 Click to open the list of hardware patterns

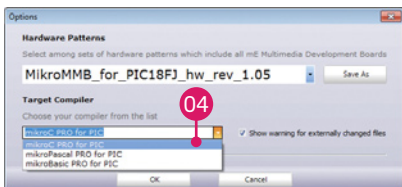
02 Select target hardware from the list



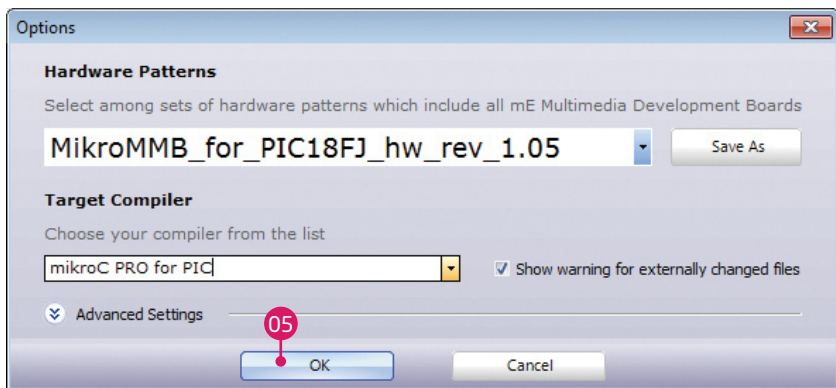
### Choose your compiler



03 Click to open a list of supported compilers

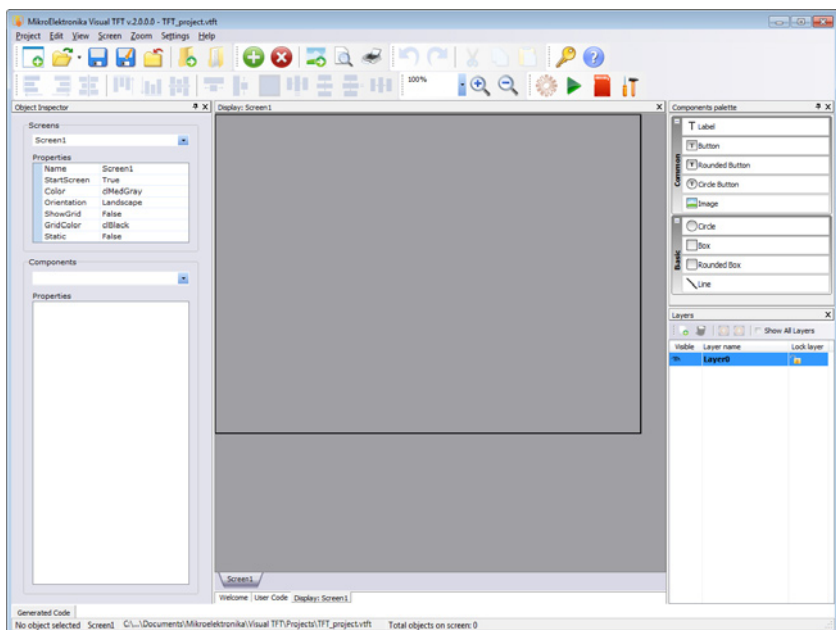


04 Select your target compiler from the list



**05** Click OK button to confirm selection of target hardware and compiler

Voila! **The first step is completed.** We have successfully created a blank new project with all the settings in place. If you followed the steps correctly you should get the window like this:





## 2. Creating visual contents

When the main options are set, the main window of the TFT display working environment will appear on the screen. Here you can create your own visual contents.

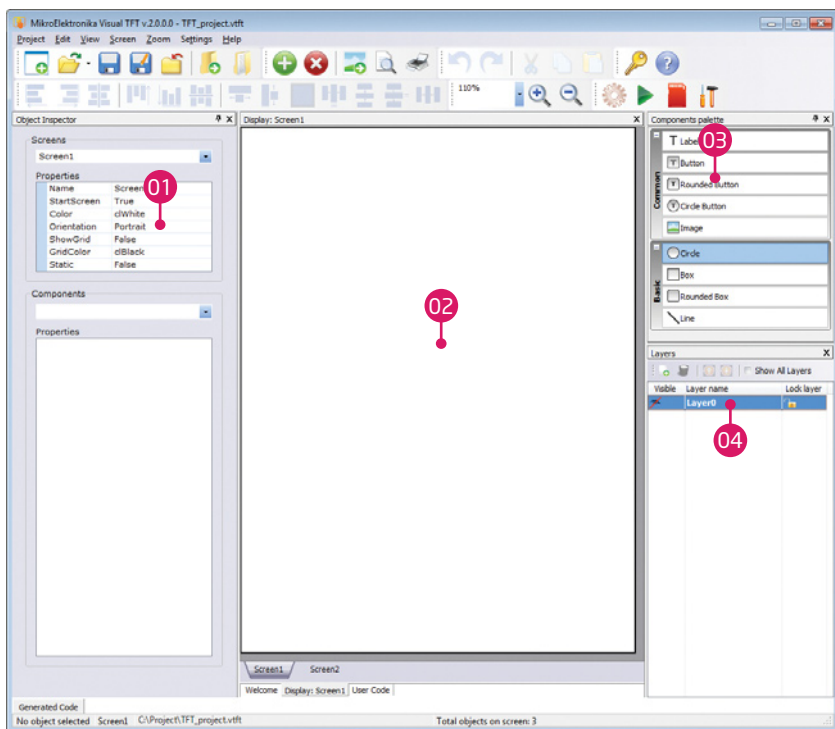


Figure 2-1: Window with a new project

- 01 In screen properties section set **Color** to **White** and **Orientation** to **Portrait**
- 02 Working section (chosen device TFT size)
- 03 Section with common and basic shapes
- 04 Layers section

With simple click and drag movements in a Visual TFT working window we will create visual content for your TFT display. In the next few steps we will show you how to do it.

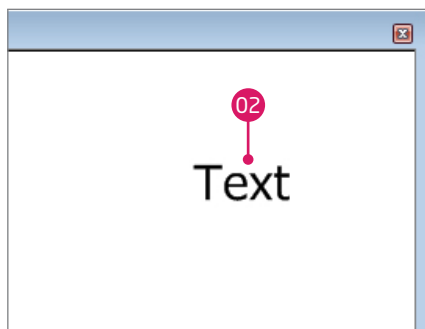
# Step 1 - Adding „Hello World“ label

When screen is set, we can now start designing the graphical user interface of our project.

## Put a Label onto the Screen

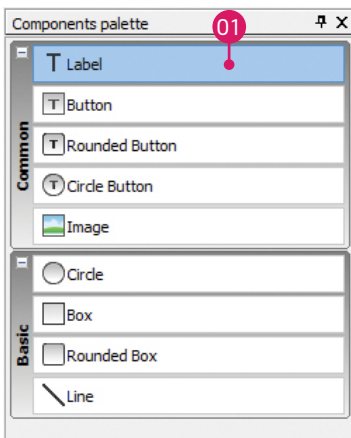
Let's add a „Hello world“ label to our project. To do that, we must place a component called **Label** from the **Components Palette** Window onto our current **Screen1**.

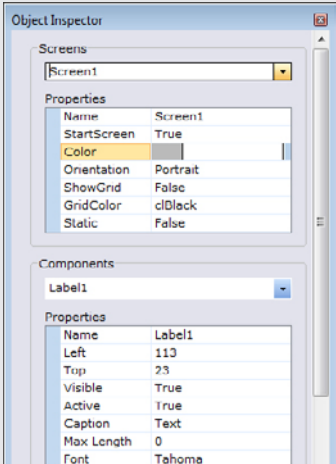
Figure 2-2: Working window with label



- 01 Click on Label and hold it
- 02 Drag it onto the Screen1 window and drop it

Figure 2-3: Components palette





Screen1	
Name	Screen1
StartScreen	True
Color	
Orientation	Portrait
ShowGrid	False
GridColor	clBlack
Static	False

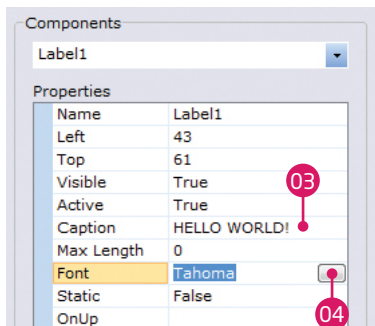
Label1	
Name	Label1
Left	113
Top	23
Visible	True
Active	True
Caption	Text
Max Length	0
Font	Tahoma

## Object Inspector

We have come to a point where we have to change component properties like text, colors and font. These settings are done in one place for all components and even all screens. As we were able to change screen color and orientation, we can also change component properties, and assign events. **Object inspector** is intuitive and clear, convenient for quick and easy changes according to your needs. It's what makes development in Visual TFT simple and rapid.

## Edit Label text, font and color

With Label component on the screen, we can now edit its properties and change text, font type, size and color in **Object Inspector**.



- 03 Click in the **Caption property** box and type in „**HELLO WORLD!**“
- 04 Click open button in Font box in order to change font settings

Figure 2-4: Components options of Object Inspector Window

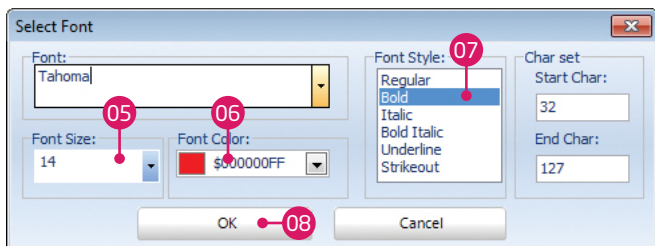
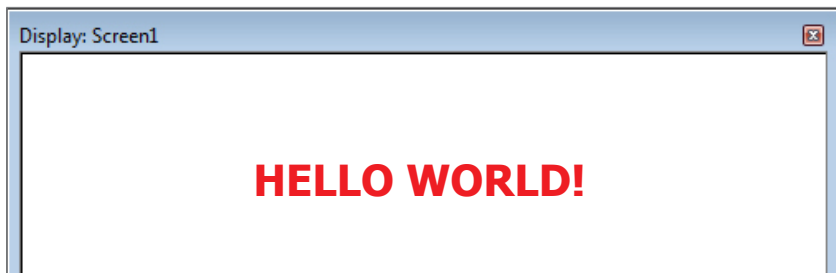


Figure 2-5: Font options

- 04 Set font size to **14**
- 05 From drop down list chose **red** color
- 06 Set font style to **Bold**
- 07 Click on **OK** button

After editing Label properties, your label should look something like this:



## Step 2 - Adding Earth image

We want to add a nice Earth picture onto our screen. Find some suitable image and convert it to bmp, jpg or jpeg format using some simple program (like Paint).

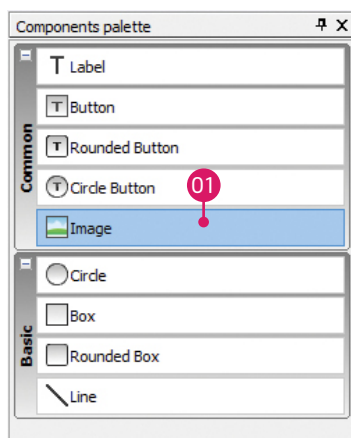
### Put an Image onto the Screen

Now, let's add an image to our project. To do that, we must place a component called **Image** from the **Components Palette** Window onto our current **Screen1**.

Figure 2-6: Working window with image



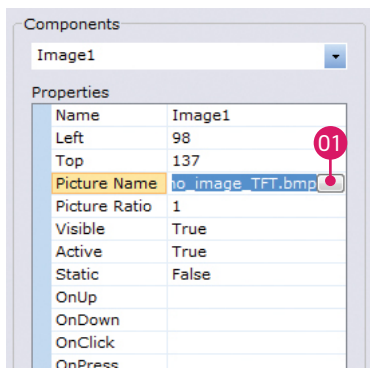
Figure 2-7: Components palette



- 01 Click on Image and hold it
- 02 Drag it in to the Screen1 window and drop it

## Assign a bitmap to the image

We can now edit Image component properties and assign bitmap we want for our project all using **Object Inspector**.



- 01 Click on the open button and new window will appear
- 02 Find image file you want to import
- 03 Select image
- 04 Click Open button

When choosing image file to your project make sure that size and format are correct. In this project we use **bmp** image format with **resolution** of 150x150 pixels.

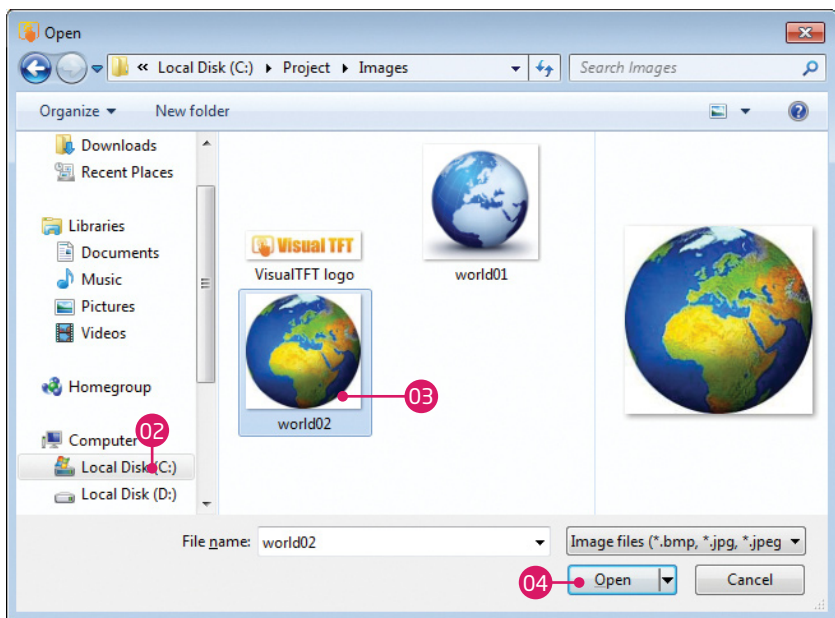


Figure 2-8: Open window

# Progress after image is added

When text and image are added your window should look like this, Figure 2-10. At this point you are just few steps away from project finish.

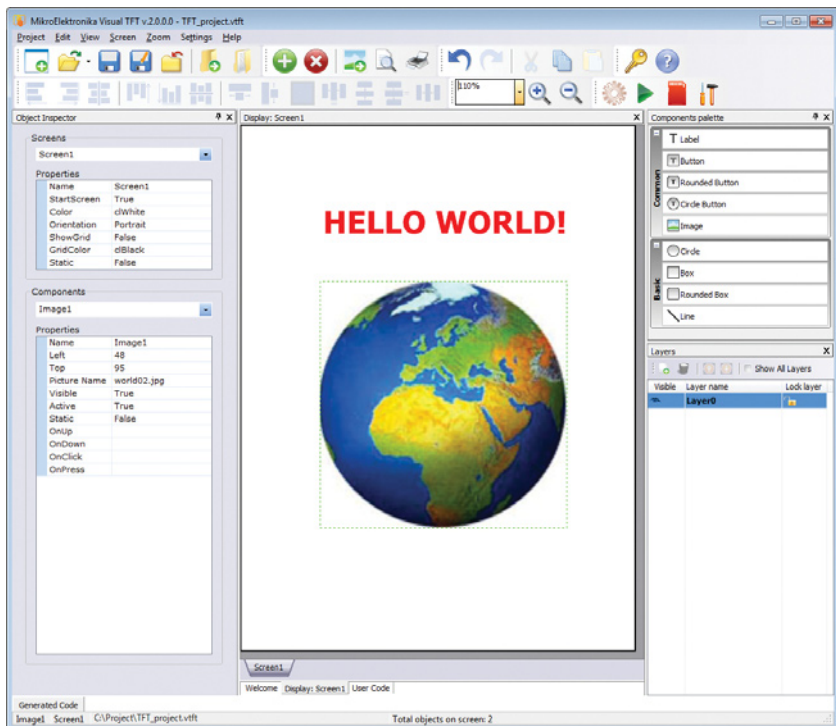


Figure 2-9: Window with added image

## Step 3 - Adding „NEXT SCREEN>“ button

To add some transition effects it is necessary to add one active component. Rounded button is perfect for this job so let's place it in the working window.

### Put Rounded Button onto the Screen

Before we proceed just add one rounded button on the screen. It will be used for OnClick transition effect which will be added later.

Figure 2-10: Components palette

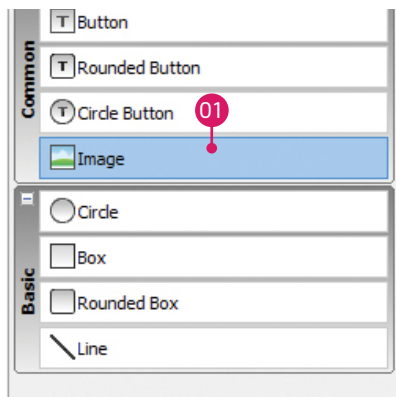
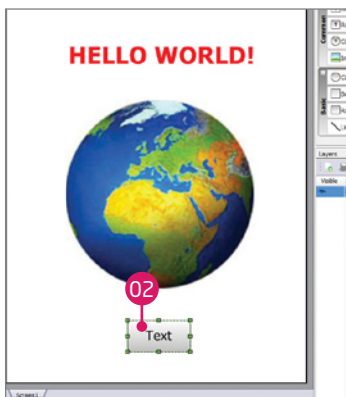


Figure 2-11: Window with button



- 01 Click the Rounded Button and hold it
- 02 Drag it in to the working section and drop it

# Renaming rounded button

Under components options chose button which will be placed in the working section:

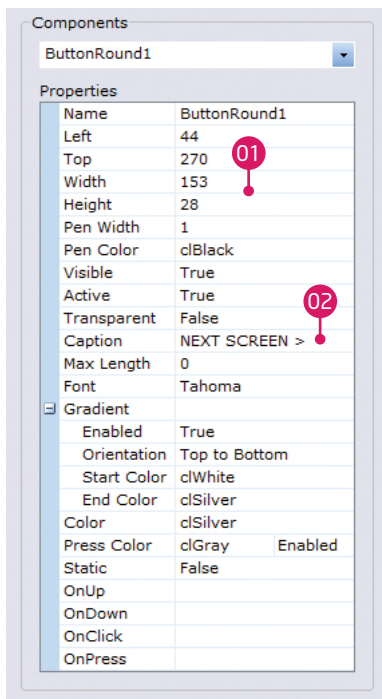


Figure 2-12: Components options

- 01 Set **Width** and **Height** to fit button text dimensions
- 02 In **Caption** box type „NEXT SCREEN >“

After rounded button is added it's time to create one more screen which will be used for transition effect. When second screen is added you can write small code which will initialize transition effect.

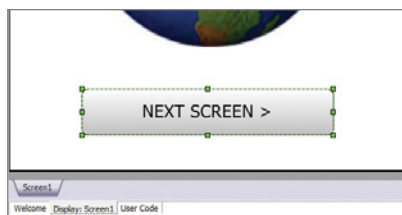


Figure 2-13: Rounded button is added



# Progress after button is added

Once the last object is placed on the working section on the main window, creation of the first screen for your TFT display is finished.

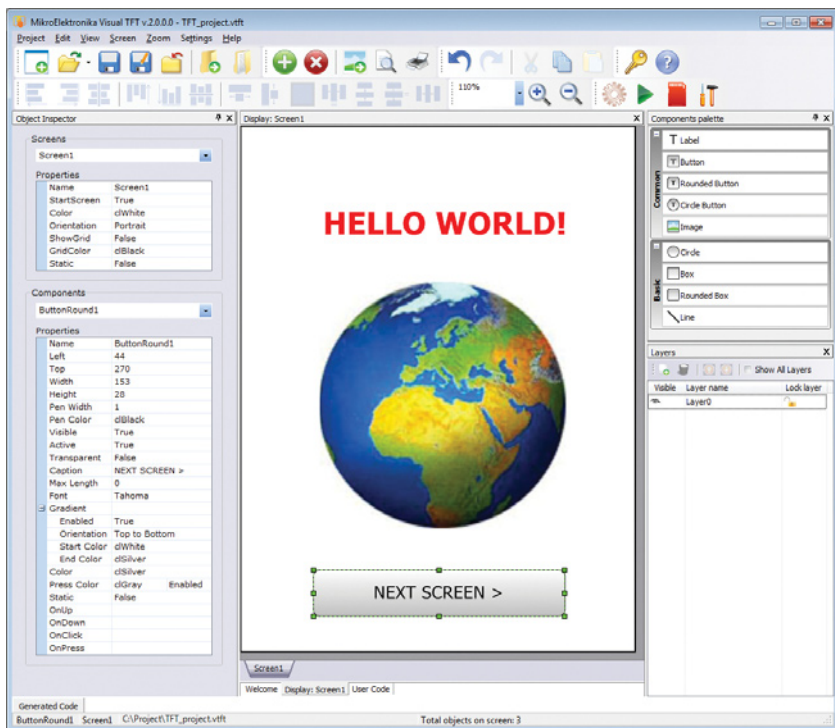


Figure 2-14: Window with added button

# 3. Creating second screen

In order to add transition effect which will occur when NEXT SCREEN > button is pressed we will add one more screen.

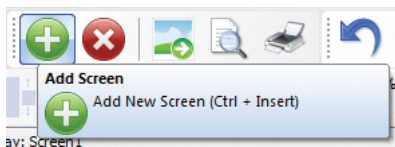


Figure 3-1: Click Add Screen button

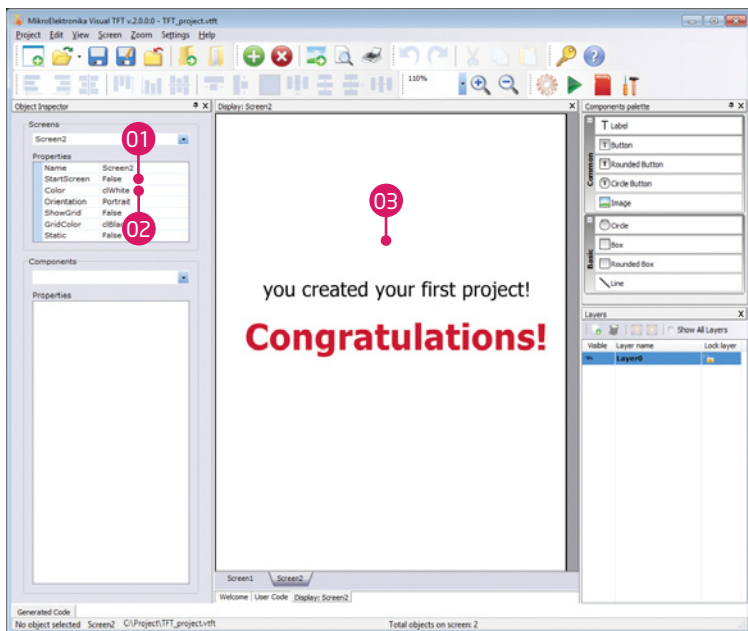


Figure 3-2: Screen2

- 01 Change background color to white
- 02 Set orientation to portrait
- 03 In order to create on screen text repeat step 1 (page 10)

## 4. Writing code

After visual part of the rounded circle is designed it's time to write user code which will draw next screen when button is clicked.

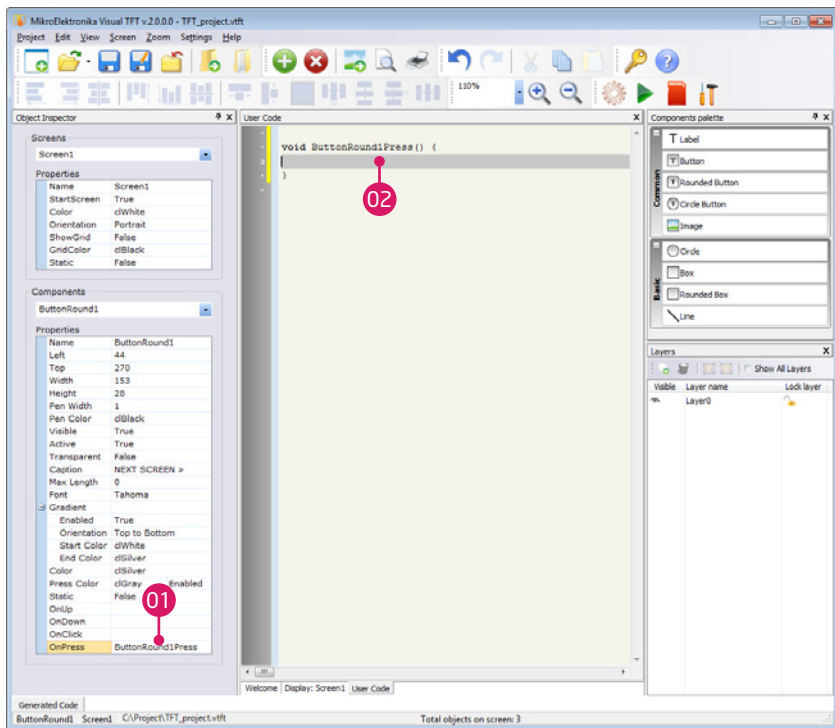
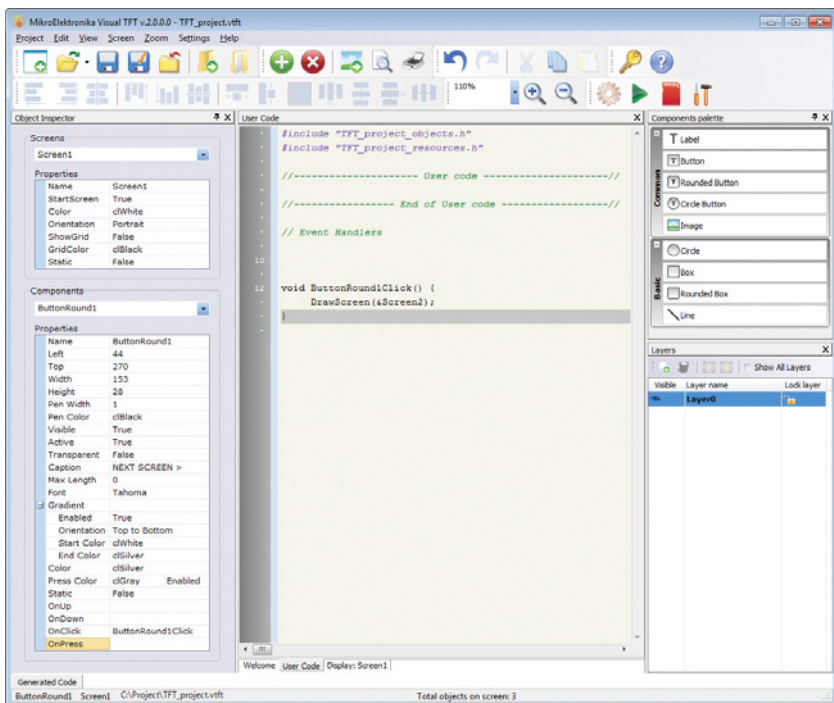


Figure 4-1: User code window

- 01 Double click in the **OnClick** event property to open user code window and type code that will be used for transition effect.
- 02 Type code between brackets

```
void ButtonRound1Click() {  
    DrawScreen(&Screen2);  
}
```

Type or copy this simple code and place it in to the user code window



**Figure 4-2: Simple code for transition effect**

In order to transfer code from Visual TFT to your device (in this case **mikromedia for PIC18FJ**) it is necessary to generate code and to start the compiler (**mikroC PRO for PIC**).

**note** Demo limit of Visual TFT is max 7 objects so this project fits in to the demo limit of the software.

*Demo limit of the compiler is 2K of program words.*

## 5. Generate code

Creation of visual content for TFT display is finished so its time to generate the code that will be transferred to the MCU memory.

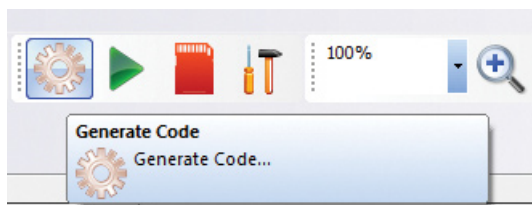


Figure 5-1: Click the Generate code button

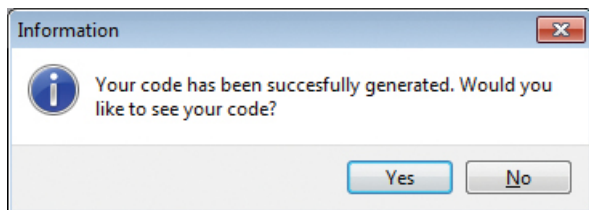


Figure 5-2: Click Yes to see generated code

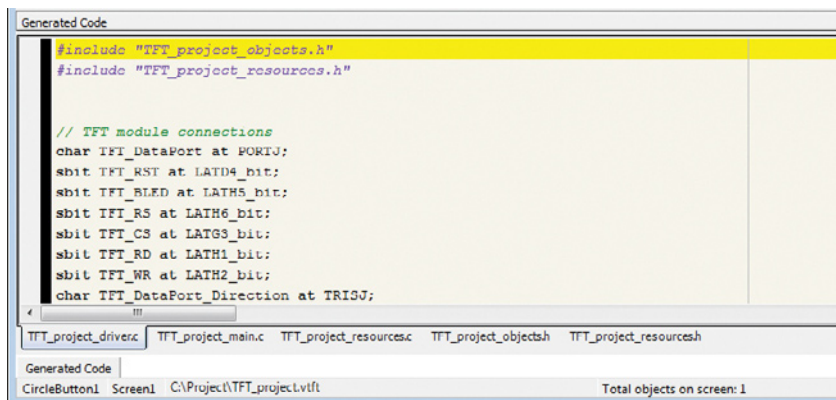


Figure 5-3: Generated code window will pop-up at the bottom of the main window

# 6. Start compiler

To transfer generated code to the device it is necessary to start the chosen compiler (in this case **mikroc PRO for PIC**).

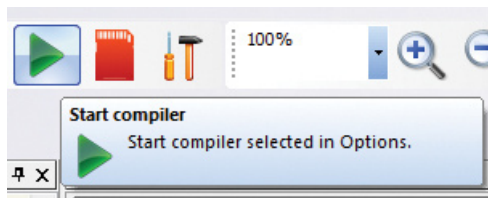


Figure 6-1: Click on Start compiler button

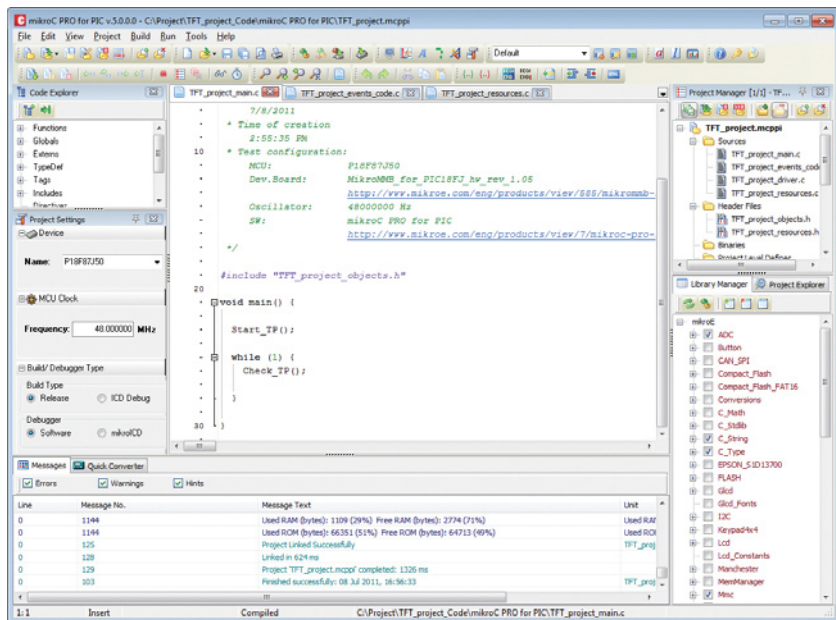


Figure 6-2: Compiler window with a code

# 7. Build and program

Attach your device to a PC via programmer (**mikroProg for PIC, dsPIC and PIC32**). In our example we use **mikromedia for PIC18FJ**.

After device is attached check settings in the compiler and then you can start build and program option in the compilers menu.

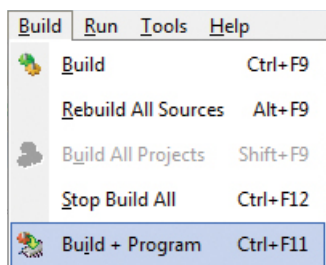


Figure 7-1: Click on the Build > Build+Program option and start code execution

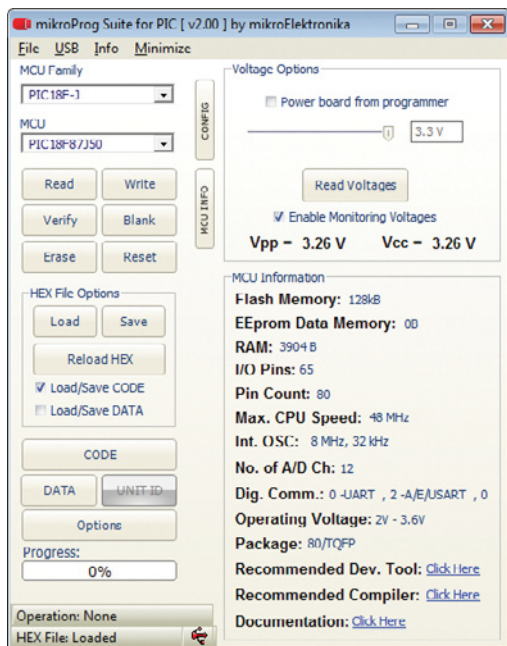


Figure 7-2: mikroProg Suite for PIC, dsPIC and PIC32 will start automatically to program device

## 8. The result

After programming process of the device is completed you can see the result of your work on the TFT display. Everything you need to do is to press **NEXT SCREEN >** button.



**Figure 8-1: Press NEXT SCREEN > button**



With one touch of the **NEXT SCREEN >** button a transition will occur. Current screen with text, image and button will be replaced with another one, Figure 8-2.



**Figure 8-2: After button is pressed a final screen will appear**

This is just the beginning of the visual creation with Visual TFT. If this project was too easy for you, you can make it much more complex with more buttons, images, lines and text, all put together in a great looking application. **Enjoy your new software!**

Notes:

## DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

## HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault - tolerant nor designed, manufactured or intended for use or resale as on - line control equipment in hazardous environments requiring fail - safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

## TRADEMARKS

The MikroElektronika name and logo, the MikroElektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVRprog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSOC4, EasyAVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of MikroElektronika. All other trademarks mentioned herein are property of their respective companies. All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

Creating first project in



If you want to learn more about our products, please visit our website at [www.mikroe.com](http://www.mikroe.com)

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at [www.mikroe.com/en/support](http://www.mikroe.com/en/support)

If you have any questions, comments or business proposals,  
do not hesitate to contact us at [office@mikroe.com](mailto:office@mikroe.com)